

DWM User's guide: Version 2.01

Y.Hao

Revised Jan 29, 2015 for FAST 8

Introduction

The Dynamic wake meandering model (DWM) is a simulator implemented in FAST, which calculates the wake deficit and meandered wake centers behind a turbine. The advantage of incorporating the DWM model into FAST is to provide the ability to obtain both the turbine power and blade loads of a downstream turbine, while maintaining an acceptable low calculation cost compared to CFD. DWM uses the rotor induction factor calculated by AeroDyn as well as some other parameters such as ambient turbulence intensity as the model input. The underlying theory behind the DWM includes the Navier Stokes equation (N-S equation) and Taylor's frozen turbulence hypothesis, which are applied to obtain the wake velocity and the meandered wake center position.

Overview of DWM implemented in FAST

The DWM model is currently constituted by two sub-models – a model of the quasi-steady wake deficit and a stochastic model of the downstream wake meandering process, as seen in Fig. 1. The DWM model is implemented in the NWTC design framework, which includes FAST, TurbSim and AeroDyn. TurbSim is used to generate random stochastic wind fields, and AeroDyn is used to calculate the aerodynamic loads on wind turbine blade elements based on the velocities and positions provided by the dynamics analysis routines from FAST. For a single turbine, the inputs of the DWM model come from two sources – the AeroDyn loads results and the DWM input text files. For a downstream turbine, the DWM inputs also include the wake deficit and the meandered wake centers of upstream turbines. The main outputs of the DWM model are the wake deficit velocity and the meandered wake center positions in the space domain behind the investigated turbine. The basic simulation flowchart is summarized in Fig. 2. More details can be found in the paper *"Implementing the Dynamic Wake Meandering Model in the NWTC Design Codes"* by Hao et al.¹

For a DWM simulation applied for more than a single turbine, the DWM-FAST program runs sequentially – for each turbine from the very upstream one to the very downstream one, the DWM-FAST runs sequentially and separately, and the outputs are written out as files for each turbine.

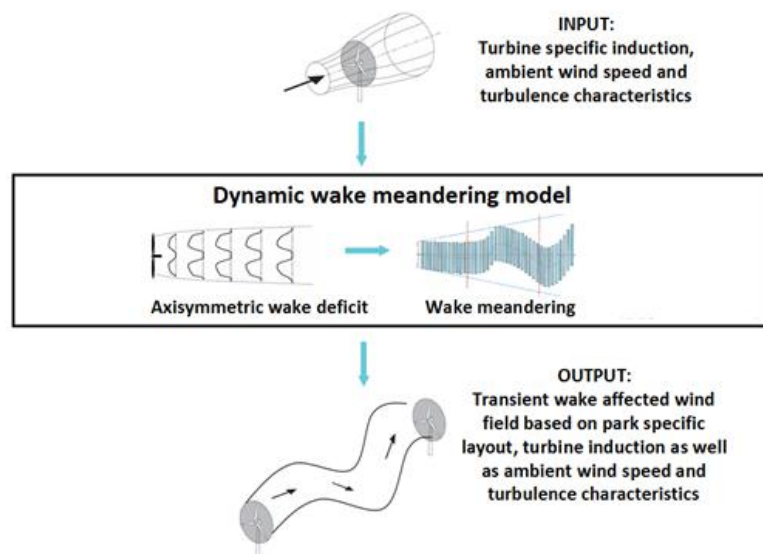


Figure 1. Overview of the DWM sub-models.²

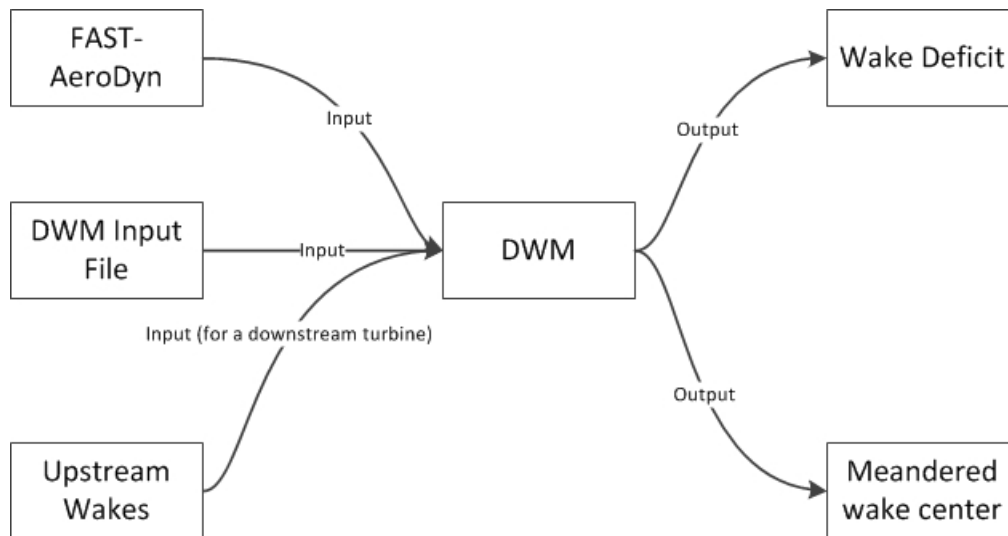


Figure 2. The flowchart of DWM implemented in FAST.

Compiling DWM

It is not necessary to compile the DWM unless you want to make changes to the code or want to run DWM using a different operating system. DWM is implemented in FAST, thus the approach of compiling DWM is the same as compiling FAST except some additional DWM files are added. The method that compiles FAST can be found in the FAST web page at

<http://wind.nrel.gov/designcodes/simulators/fast/alpha/>

Using DWM

To run the DWM executable program, the same approach that runs FAST can be applied here. All DWM output files have the specified root file name. The method that runs FAST can be found in the FAST web page at

<http://wind.nrel.gov/designcodes/simulators/fast/alpha/>

However since the DWM runs sequentially for a wind farm, a **driver program** has been built to manage the series of DWM-FAST simulations. If to run a single DWM-FAST to estimate the wake, the driver program **must** be run for the case “0” (in the debug model). The running of the “0” case is to pre-screen the wind farm and read in the model parameters, and outputs a binary file which will be read in by the DWM-FAST program.

To perform a series of DWM simulations, instead of *DWM.exe*, the *DWM_Driver.exe* should be run (the DWM driver program will call a sequence of DWM-FAST to run). The *DWM.exe* as well as the *DWM_Driver.exe* should both be put under the same directory location where the FAST input file is located. The input file for the DWM-FAST should be put in the sub-directory */DWM-driver* and the DWM outputs files will be written out under the sub-directory */DWM-results* when the DWM

simulation finishes.

More information related to the compiling and running of the DWM driver program and the DWM-FAST are provided in the “DWM-FAST compiling introduction”.

To run DWM_Driver.exe, open a command prompt window in the directory in which you want to work. The command-line syntax is:

```
DWM_Driver.exe <FAST input file>
```

An example of the DWM-FAST driver program command line input is shown in Fig.3.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>d:

D:\>cd U80_FAST8

D:\U80_FAST8>DWM_driver.exe FAST_U80
```

Figure 3. Example DWM-Driver program command line input.

Distributed Files

Currently, the DWM is treated as a sub-module of AeroDyn such that some modifications are made in the AeroDyn module. The archive contains the DWM-Driver program and DWM program implemented in FAST8. See Table 1 and Table 2 for a complete list of the files included in the DWM-Driver program and the DWM-FAST archive (excluding the FAST files) respectively.

DWM-Driver Source File	Description
<i>DWM_driver.f90</i>	DWM-Driver program main file
<i>DWM_driver_data.f90</i>	Contains the DWM-Driver data
<i>DWM_driver_sub.f90</i>	Contains the subroutines that used by DWM-Driver

Table 1. DWM-Driver program source file.

DWM Source File	Description
-----------------	-------------

<i>DWM.f90</i>	DWM module main file
<i>DWM_Wake_Sub.f90</i>	Contains the subroutines that used by DWM
<i>DWM_Types.f90</i>	DWM data types
<i>Registry-DWM.txt</i>	Registry file for DWM

Table 2. DWM program source file.

Input File

DWM reads a text input file to set the parameters required for the program to execute. Do not change the name of the text input file unless corresponding modifications are done in the driver program source file. DWM assumes that parameters are located on specific lines, so do not add or remove lines from the sample input files included in the archive except the turbine coordinate section. The heading for each parameter contains its name, a description, and the units for that parameter. The names of the parameters are provided for reference, and DWM does not read those names from the input file. An example DWM input file is included in Appendix A. The DWM input parameters are:

HubHt: The turbine hub height [m]

This is the turbine hub height, the value of this parameter cannot be negative.

RotorR: The radius of rotor blade [m]

This parameter specifies the distance from the rotor apex to the blade tip. The value of this parameter cannot be negative.

NumWt: The total number of wind turbines [-]

This parameter specifies the total number of the wind turbine that to be simulated. The value of this parameter cannot be smaller than 1.

Uambient: The ambient wind velocity [m/s]

This is the mean wind speed at the turbine hub height. The value of this parameter cannot be negative.

TI: The ambient turbulence intensity [%]

This is the average ambient turbulence intensity in percentage at the hub height or the TI of the TurbSim generated wind file. The value of this parameter cannot be negative.

ppR: The number of points per radius [-]

DWM uses the finite difference scheme to solve the fluid dynamic system. This parameter defines the number of calculation node in the spanwise direction. In general, larger nodal values cost more calculation time but yield more precise results. The value of this parameter must be an integer and 50 is recommended.

Domain_R: The radial domain size [R]

DWM applies an axisymmetric coordinate system to solve the fluid dynamic system. This parameter specifies the radial distance from the center of the calculation domain to the very boundary edge and

is scaled by the rotor radius. The value of this parameter cannot be smaller than 3 and should be adjusted based on the turbine layout if there are more than single turbines in this simulation. If the wind direction is not down the row, then the radial domain size should be larger than $spacing * \sin(\theta)$ where θ is the turbine alignment angle

Domain_X: The longitudinal domain size [R]

This is the longitudinal domain size counted from the investigated rotor plane. The value of this parameter must be positive and should be determined based on the turbine layout if there is more than a single turbine in this simulation. For example, this value should be larger than the maximum spacing between two neighboring turbines if a downstream turbine is only affected by the closest upwind turbine.

Meandering_simulation_time_length: The total number of simulation time steps in the meandering wake model for a fixed cross plane [-]

Meandering_Moving_time: The total number of simulation time steps in the meandering wake model for a moving cross plane [-]

These two parameters control the wake meandering sub-model. Taylor's frozen turbulence hypothesis is applied for the downstream advection of the wake, and the fundamental assumption of this approach is that the wake transport in the atmospheric boundary layer can be modeled, by considering the wake to act as a passive tracer driven by large scale turbulence.³ To calculate the wake displacement in the vertical and lateral directions, the wake is modeled as constituted by a cascade of wake deficits, each "emitted" at consecutive equally spaced time increments, in agreement with the passive tracer analogy.

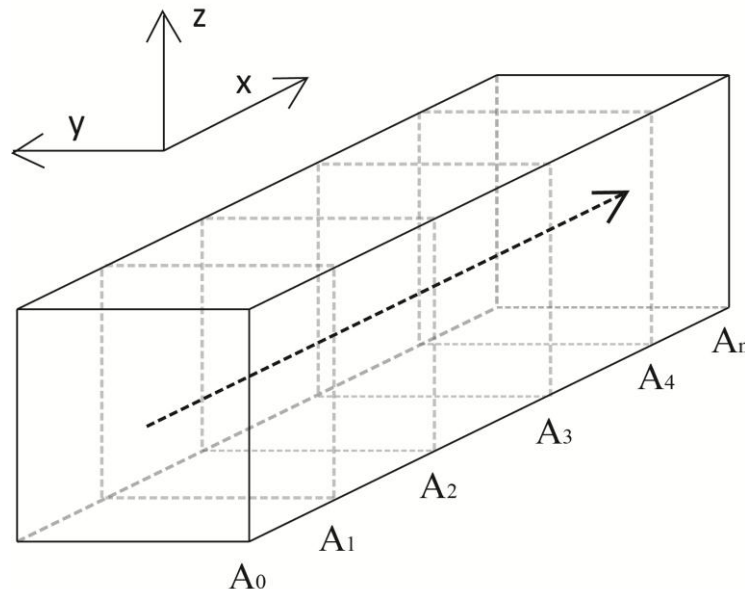


Figure 4. Illustration of the wake meandering model space domain.

At every time step, a cross-plane is released from the rotor plane and marches downstream while the wind properties of this cross-plane are constant. The entire space domain behind the turbine consists

of multiple cross-planes, which are all released from the rotor plane with the spacing between two neighboring cross-planes equal to the ambient velocity multiplied by the time step interval. In each cross-plane there is a wake center position. For a single cross-plane, it marches a constant distance downstream at every time step, and a new wake center position is obtained using a filter function, with the averaged vertical and lateral velocity calculated based on the wake center position of the last time step.

As illustrated in Fig. 4, the wake meandering model space domain behind the turbine is a cuboid, with the x direction in the mean flow nominal direction, the y direction in the lateral direction, the z direction in the vertical direction, the investigated turbine located at the cross plane location A_0 , and the very downstream boundary plane at A_n . The wake meandering model functions as follows.

First, at time t_0 , a frozen cross plane C_0 is released from the turbine plane A_0 and starts to advect downstream. After time Δt , this frozen cross plane C_0 advects to the location of the cross plane A_1 . Finally at time T this frozen cross plane C_0 reaches the location where the cross plane A_n locates. Throughout this whole process at each time step and at each cross plane location A , a meandered wake center vertical and lateral position coordinate in the cross plane C_0 is returned, and the wake center coordinate at the x direction is equal to the ambient velocity multiplied by the local total time which is counted from the original release. A cross plane C is released at every time step and the same approach discussed above is applied for each cross plane.

The parameter *Meandering_Moving_time* specifies the total time T for which a frozen cross plane travels, and the parameter *Meandering_simulation_time_length* specifies the total number of the cross planes that advect from the turbine cross plane A_0 to the very downstream boundary plane A_n .

The values of the two parameters both must be positive integers. And to obtain the appropriate parameters values, Eq. (1) and Eq. (2) are shown below.

$$MSTL \geq Ztime / \left(\frac{\frac{2R}{ppR}}{0.32U_{ambient}} \cdot 10 \right) + 1 \quad (1)$$

$$MMT \geq spacing \cdot \frac{ppR}{10} + 1 \quad (2)$$

MSTL is the parameter *Meandering_simulation_time_length*, *MMT* is the parameter *Meandering_Moving_time*, *Ztime* is the parameter *Tmax* specified in the FAST input file which is the total run time (s), and *spacing* is the distance normalized by rotor diameter from the downstream investigated turbine to the very upstream turbine whose wake will affect this downstream turbine

WFLowerBd: The lower bound of the wind file [m]

This parameter specifies the lower bound of the generated inflow wind file.

Winddir: The ambient wind direction [degree °]

This parameter specifies the incoming ambient wind direction in units of degrees. The value of this parameter must be between 0 and 360. The degree measurements are top-wise as shown in Fig. 5 below. If looking down on the wind farm, 0° means the wind comes from the very top or north, 90° the very right or east, 180° the very bottom or south and 270° the very left or west.

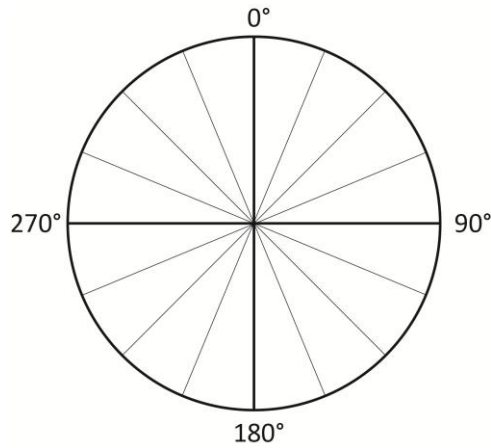


Figure 5. Illustration of wind direction degree measurements.

DWM.exe name: The file rootname of the DWM-FAST program

This is the root name of the DWM-FAST executable file.

XCoordinate, YCoordinate: The coordinate of turbine [D]

These parameters are the coordinates of the turbines at x and y directions in the Cartesian coordinate system, and they are scaled by the rotor diameter. The total number of the lines must be equal to the total number of the turbines. At each line, the x coordinate is typed in first and it is followed by the y coordinate of the same turbine. Based on the order of the turbine coordinate inputs, the turbines are sorted numerically from 1 to the total number of turbines.

Output Files

Besides the normal FAST output files that record the power and loads results, several text files are created by DWM as the output files. They have the root name of the turbine index and some keywords indicating what kind of results are stored in these text files.

FAST .out and .elm output files

Since the DWM runs sequentially for each turbine, if there are more than a single turbine, the FAST .out and .elm output files are renamed corresponding to the investigated turbine index. For example, the *FastOutput_Turbine_2.out* and *FastElm_Turbine_2.elm* stand for the FAST .out and FAST .elm output files for turbine 2 respectively.

Induction factor output files

The induction factor describes the amount of the free stream wind speed is reduced by the wind turbine, and is one of the inputs for the DWM model. However, the axial induction factors given by AeroDyn must be interpreted carefully, as the tip loss correction in AeroDyn causes the calculated induction values near the tip to be too large and unphysical. To calculate the axial induction factors for DWM, which reflect the exchange of momentum between the fluid and the rotor, the induction factor a is generated directly from the thrust coefficient C_t , which is related to the thrust force F , the swept area A , the air density ρ and the wind speed U . The equations reflecting the above relationships for the i^{th} annulus are shown in Eq. (3) and Eq. (4).

$$Ct_i = \frac{F_i}{\left(\frac{1}{2}\rho A_i U^2\right)} \quad (3)$$

$$Ct_i = 4a_i(1 - a_i) \quad (4)$$

The number of induction factor values in the induction factor output files is equal to the number of blade nodes used for the AeroDyn analysis. The name of the induction factor output files are named corresponding to the investigated turbine index. For example, *Induction_Turbine_2.txt* stores the induction factors for the turbine 2.

Mean velocity output files

The mean velocity in *m/s* is the time and spatial averaged velocity for a turbine throughout the whole simulation time domain. For example, *Mean_U_Turbine_2.txt* store the mean velocity for the turbine 2, and the total number of these mean velocity output files is equal to the total number of the turbines simulated.

Turbulence intensity output files

The turbulence intensity (TI) reflects the intensity of the mixing between the wake and ambient flow. If a turbine is located in the wake of an upstream turbine, the turbulence intensity at the downwind turbine is no longer equal to the ambient turbulence intensity. Instead, the downstream turbine's turbulence intensity combines the wake added turbulence and the apparent turbulence intensity which is due to the large scale wake meandering both from the upwind turbines. Each turbine has its own turbulence intensity output file and the TI is scaled at the rotor plane. For example, *TI_Turbine_2.txt* stores the turbulence intensity at the turbine 2.

Wake deficit velocity output files

When the wind flows past a wind turbine, the extraction of the axial momentum in the wind causes the wake deficit behind a turbine. The wake deficit velocity is one of the major outputs that DWM calculates. Since the DWM uses the finite difference scheme to solve the fluid system, the velocity values are obtained at each calculation nodal position in the space domain, whose size is defined by the *ppR*, *DomainR* and *DomainX* in the DWM input file. The values of the wake velocity are between 0 and 1 since they are normalized by the ambient wind velocity. These wake velocity files are renamed corresponding to the investigated turbine index. For example, *WakeU_Turbine_2.txt* stores the wake velocity behind the turbine 2. A MATLAB script file is included in this archive to post-process the wake deficit output, which generates the normalized wake velocity at a certain downstream plane as Fig. 6 presents below.

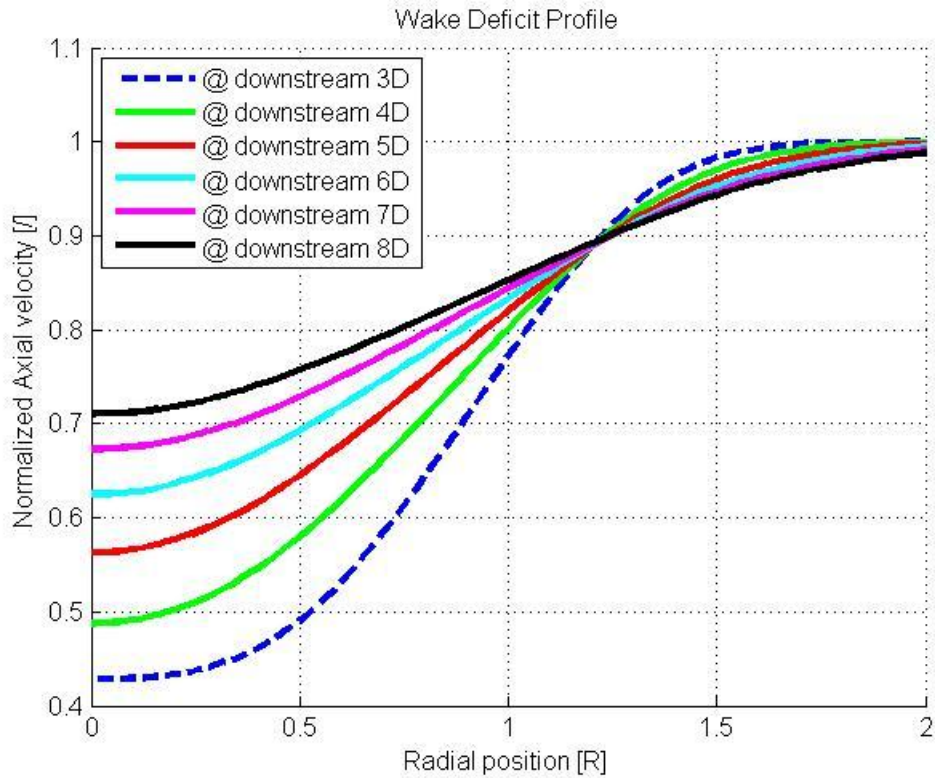


Figure 6. Example wake deficit profiles at various downstream locations plot.

Meandered wake center output files

Wake meandering is the term used to describe the large scale movement of the entire wake. Wake meandering is important because it might considerably increase extreme loads and fatigue loads, in particular yaw loads, on turbines downstream in wind farms, because the meandering causes the wake to be swept in and out of the rotor plane of downstream turbines. The wake center positions are expressed in the Cartesian coordinate system in x-y-z directions which is the same as the tower-base coordinate system used by the FAST. The meandered wake center output files are renamed corresponding to the investigated turbine index. For example, *WC_Turbine_2.txt* records the meandered wake center coordinates behind turbine 2 for all the simulation time steps. A MATLAB script file is included in this archive to post-process the meandered wake center coordinates output, which generates the meandered wake center coordinates at a certain time step. An example of the wake center coordinate in the vertical direction and lateral direction for different moving times are shown in Fig.7 and Fig.8 respectively.

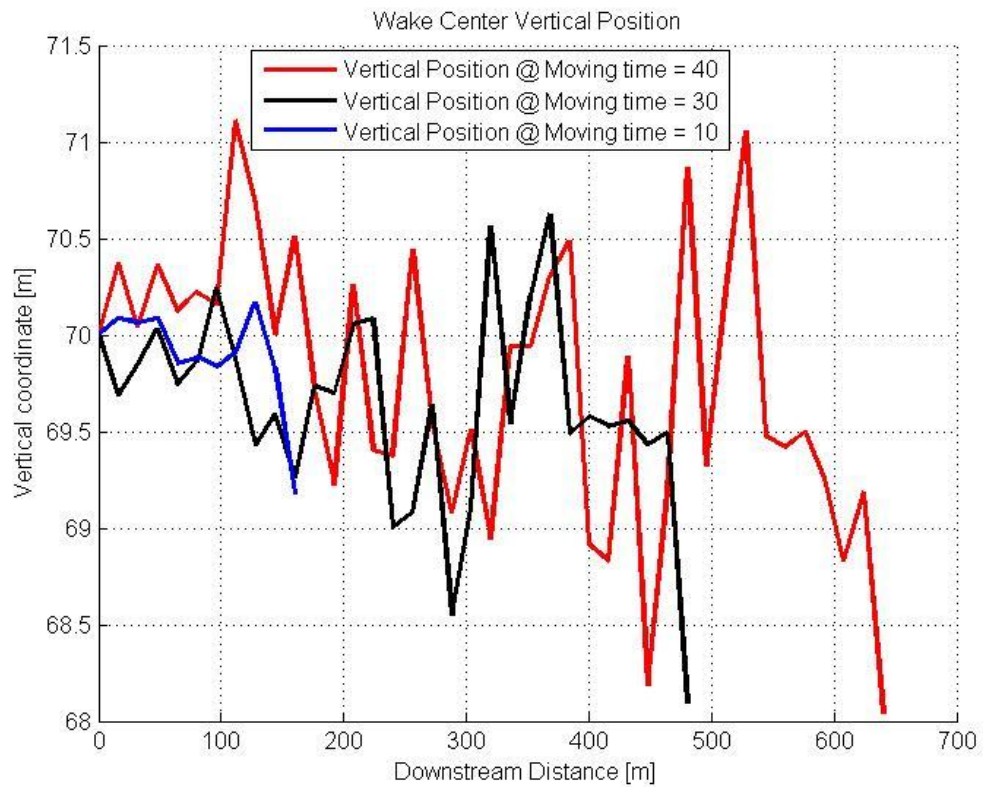


Figure 7. Example meandered wake center coordinates in vertical direction plot.

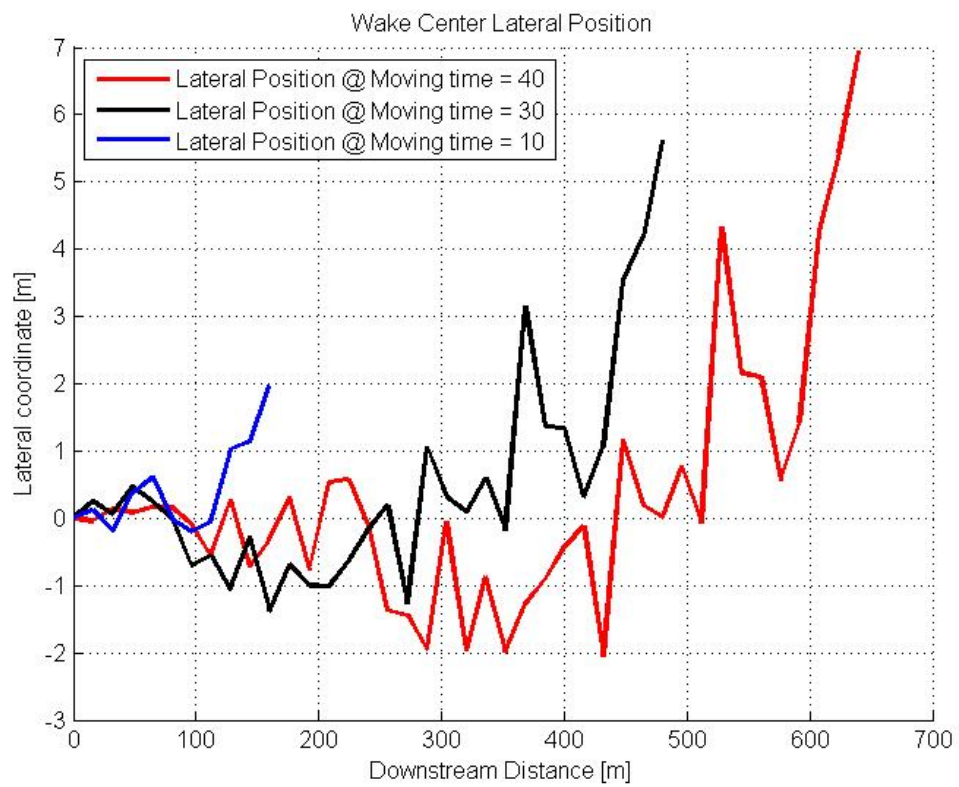


Figure 8. Example meandered wake center coordinates in lateral direction plot.

References

¹Hao Y, Lackner M, Keck R, Lee S, Churchfield M, Moriarty P, “Implementing the Dynamic Wake Meandering Model in the NWTTC Design Codes,” Dec 2013.

²Keck R, Veldkamp D, Madsen H, Larsen G, “Implementation of a Mixing Length Turbulence Formulation Into the Dynamic Wake Meandering Model,” *Journal of Solar Energy Engineering*, 134(2):021012-021012-13, 2012.

³Gunner C. Larsen, Helge Aa. Madsen, Ferhat Bingöl, Jakob Mann et al., “Dynamic wake meandering Modeling,” Risø-R-1607(EN), June 2007.

Appendix A: Sample DWM Input File

```
----- DWM WIND FARM INPUT FILE -----
DWM wind farm specification.
Implemented in the FAST8 with DWM module
----- Wind Farm Parameters -----
70  HubHt          - The hub height (m)
40  RoterR         - The Rotor radius (m)
4   NumWT          - The total number of wind turbines (-)
7.9 Uambient       - The ambient wind velocity (m/s)
6   TI             - Ambient turbulence intensity (%)
50  ppR            - The number of points per radius (-)
5.0 Domain_R       - Radial domain size (R) (should be larger than spacing*sin(theta) where theta is the turbine alignment angle)
36  Domain_X       - Longitudinal domain size (R) (should be larger than turbine spacing)
300 Meandering_simulation_time_length - The total simulation time steps in the meandering wake model for a fixed cross plane (-)
60  Meandering_Moving_time - The total simulation time steps in the meandering wake model for a moving cross plane(-)
205 TurbSimHubHt    - The TurbSim wind file reference height (m)
180 Winddir        - The ambient wind direction (degree)
XCoordinate      YCoordinate
0                11
0                22
0                33
0                44
```